

IN THE CLAIMS

Please amend claims 6, 18 and 20 as indicated below.

This listing of claims will replace all prior versions, and listings, of the claims in the Application:

**Listing of Claims:**

1 Claim 1 (original) A method of generating a global history vector comprising the  
2 steps of:

3 determining if a selected group of instructions contains a branch instruction;  
4 maintaining a current global history vector in a shift register when the selected  
5 group does not contain a branch instruction;

6 shifting a first value into the shift register to generate a second vector if the  
7 selected group contains a branch instruction and the branch instruction is predicted as  
8 a branch taken; and

9 shifting a second value into the shift register to generate a second vector when  
10 the selected group contains a branch instruction and the selected group does not  
11 include a branch instruction predicted as a branch taken.

1 Claim 2 (original) The method of Claim 1 and further comprising the step of storing  
2 the generated value in an entry in a branch instruction queue associated with the  
3 selected group of instructions.

1 Claim 3 (original) The method of Claim 2 and further comprising the step of  
2 correcting the generated vector upon a misprediction comprising the substeps of:

3 retrieving a selected number of bits of the vector stored from the branch  
4 instruction queue into the shift register; and

5 shifting an updated history bit into the shift register.

1 Claim 4 (original) The method of Claim 1 wherein the first value comprises a logic 1  
2 and the second value is a logic 0.

1 Claim 5 (original) The method of Claim 1 wherein the selected group of instructions  
2 comprises eight instructions.

1 Claim 6 (currently amended) A method of performing branch predictions comprising  
2 the steps of:

3 indexing a branch history table using a first global history vector associated  
4 with a first fetch group of instructions during a first fetch cycle to retrieve a first  
5 prediction value;

6 generating a second global history vector associated with a second fetch group  
7 of instructions comprising the substeps of:

8 retaining the first vector when the first fetch group does not contain at  
9 least one branch instruction;

10 appending a bit of a first value to the first vector when the first fetch  
11 group has at least one branch instruction predicted to be a branch taken; [[and]]

12 appending a bit of a second value to the first vector when the first  
13 group contains at least one branch instruction and contains no branch instructions  
14 predicted to be a branch taken; and

15 indexing the branch history table using the second global history  
16 vector during a second fetch cycle to retrieve a second branch prediction value.

1 Claim 7 (previously presented) The method of Claim 6 and further comprising the  
2 step of storing the first and second vectors in an entry of a branch history queue  
3 associated with the first fetch group.

1 Claim 8 (original) The method of Claim 7 and further comprising the steps of:

2 detecting a branch misprediction based on the first prediction value;

3 retrieving the first and second vectors from the branch history queue;

4 indexing the branch history table using the first vector to correct the first  
5 prediction value; and

6 appending a corrected bit to the second vector to generate a corrected branch  
7 history vector.

1 Claim 9 (original) The method of Claim 7 wherein said first fetch cycle precedes the  
2 second fetch cycle by three fetch cycles.

1 Claim 10 (original) The method of Claim 7 wherein said steps of indexing comprises  
2 the step of gating the vector with selected bits of a current instruction address.

1 Claim 11 (original) The method of Claim 10 wherein said steps of gating comprise  
2 the steps of performing XOR operations.

1 Claim 12 (previously presented) The method of Claim 8 wherein said substeps of  
2 appending comprise the substeps of shifting a bit into a shift register storing the  
3 second vector.

1 Claim 13 (original) Branch processing circuitry comprising:

2 a shift register for storing a global history vector;

3 control circuitry for selectively updating a first global history vector stored in  
4 said shift register operable to:

5 determine if a selected group of instructions contains a branch  
6 instruction;

7 maintain said first global history vector in said shift register when the  
8 selected group does not contain a branch instruction;

9 shift a first value into the shift register to generate a second vector if  
10 the selected group contains a branch instruction and the branch instruction is  
11 predicted as a branch taken; and

12 shifting a second value into the shift register to generate a second  
13 vector when the selected group contains a branch instruction and does not contain a  
14 branch instruction predicted as a branch taken.

1 Claim 14 (original) The branch processing circuitry of Claim 13 and further  
2 comprising a branch history table and circuitry for generating an index to an entry in  
3 said branch history table using selected bits from a current address and selected bits  
4 of said first vector to retrieve a prediction value stored therein.

1 Claim 15 (original) The branch processing circuitry of Claim 14 and further  
2 comprising circuitry for updating said second vector when said prediction value  
3 results in a misprediction comprising:

4 a queue for storing said first and said second vectors;  
5 circuitry for accessing said vectors from said queue;  
6 circuitry for indexing said branch history table with said first vector and  
7 updating a corresponding entry with a corrected prediction value; and  
8 circuitry for updating a vector in said shift register with said second vector;  
9 and  
10 circuitry for shifting the corrected prediction value into said shift register.

1 Claim 16 (original) The branch processing circuitry of Claim 13 wherein said branch  
2 processing circuitry forms a portion of a single-chip microprocessor.

1 Claim 17 (original) A processing system comprising:

2 a microprocessor comprising:  
3 a branch history table for storing branch prediction values;  
4 a global history shift register for storing a global branch history vector;  
5 logic for generating an index to said branch history table and accessing  
6 prediction values stored therein using selected bits of a said branch history vector  
7 stored in said shift register; and  
8 control circuitry for updating a said global branch history vector stored  
9 in said shift register and operable to:  
10 retain a current vector stored in said shift register when a  
11 selected fetch group does not contain at least one branch instruction;  
12 shift a bit of a first value into said shift register to generate an  
13 updated vector when the selected fetch group has at least one branch instruction  
14 predicted to be a branch taken; and  
15 shift a bit of a second value into said shift register when said  
16 selected fetch group contains at least one branch instruction and contains no branch  
17 instructions predicted to be a branch taken.

1 Claim 18 (currently amended) The processing system of Claim 17 wherein said  
2 microprocessor further comprises:

3 a branch instruction queue having a plurality of entries each associated with a  
4 [[said]] fetch group for storing at least first and second corresponding global history  
5 vectors;

6 circuitry for detecting a misprediction associated with a [[said]] prediction  
7 value retrieved from said branch history table and corresponding to said first global  
8 history vector in said branch instruction queue;

9 circuitry for retrieving said first vector from said branch instruction queue and  
10 accessing a corresponding entry in said branch history table to correct said prediction  
11 value stored therein; and

12 circuitry for retrieving and modifying said second vector to generate a  
13 corrected vector in said shift register.

1 Claim 19 (original) The processing system of Claim 17 wherein said processing  
2 system further includes a system memory coupled to said microprocessor by a bus.

1 Claim 20 (currently amended) The processing system of Claim 17 wherein [[a]] said  
2 fetch group comprises eight instructions.